



ASW

THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Applicant: Oleg Rashkovskiy, et al.	§	Art Unit: 2131
	§	
Serial No.: 09/706,501	§	Examiner: Arezoo Sherkat
	§	
Filed: November 2, 2000	§	Atty Docket: ITL.0778US
	§	(P10142)
For: Content Protection Using Block	§	
Reordering	§	Assignee: Intel Corporation
	§	
Cust. No.: 21906	§	Conf. No.: 8091

Mail Stop **Appeal Brief**  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

TRANSMITTAL OF AMENDED APPEAL BRIEF

Dear Sir:

In response to the Notification of Non-Compliant Appeal Brief, attached hereto is an Amended Appeal Brief.

Headings in the Grounds of Rejection and Arguments sections have been amended to show grounds of rejection as indicated in the final Office action. The Amended Appeal Brief is therefore believed to be in compliance.

No fee is believed to be due with this response. However, the Commissioner is authorized to charge any fee due to Deposit Account No. 20-1504 (ITL.0778US).

Respectfully submitted,

Date: September 20, 2006

Rhonda L. Sheldon, Reg. No. 50,457  
TROP, PRUNER & HU, P.C.  
1616 S. Voss Road, Suite 750  
Houston, TX 77057  
(713) 468-8880 [Phone]  
(713) 468-8883 [Fax]

Date of Deposit: September 20, 2006  
I hereby certify under 37 CFR 1.8(a) that this correspondence is being deposited with the United States Postal Service as **first class mail** with sufficient postage on the date indicated above and is addressed to Mail Stop Amendment, Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450.  
  
Nancy Meshkoff



Conf. No.: 8091

  
Nancy Meshkoff

## **TABLE OF CONTENTS**

REAL PARTY IN INTEREST.....	3
RELATED APPEALS AND INTERFERENCES.....	4
STATUS OF CLAIMS .....	5
STATUS OF AMENDMENTS .....	6
SUMMARY OF CLAIMED SUBJECT MATTER .....	7
GROUND OF REJECTION TO BE REVIEWED ON APPEAL .....	10
ARGUMENT .....	11
CLAIMS APPENDIX.....	22
EVIDENCE APPENDIX.....	30
RELATED PROCEEDINGS APPENDIX.....	31

**REAL PARTY IN INTEREST**

The real party in interest is the assignee Intel Corporation.

**RELATED APPEALS AND INTERFERENCES**

None.

## **STATUS OF CLAIMS**

Claims 1-28 (Rejected).

Claims 29-55 (Withdrawn).

Claims 56-76 (Rejected).

Claims 77-78 (Withdrawn).

Claims 79-81 (Rejected).

Claims 82-90 (Withdrawn).

Claims 91-96 (Rejected).

Claims 1-28, 56-76, 79-81, and 91-96 are rejected and are the subject of this Appeal Brief.

## **STATUS OF AMENDMENTS**

All amendments have been entered.

## SUMMARY OF CLAIMED SUBJECT MATTER

In the following discussion, the independent claims are read on one of many possible embodiments without limiting the claims:

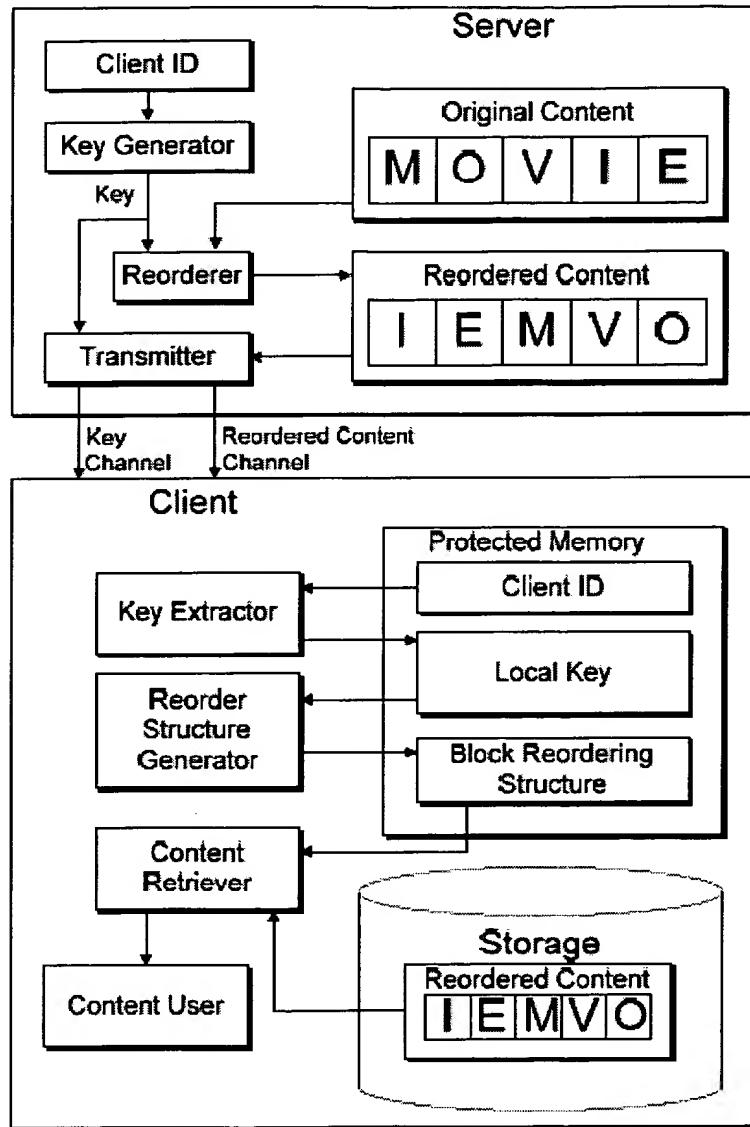


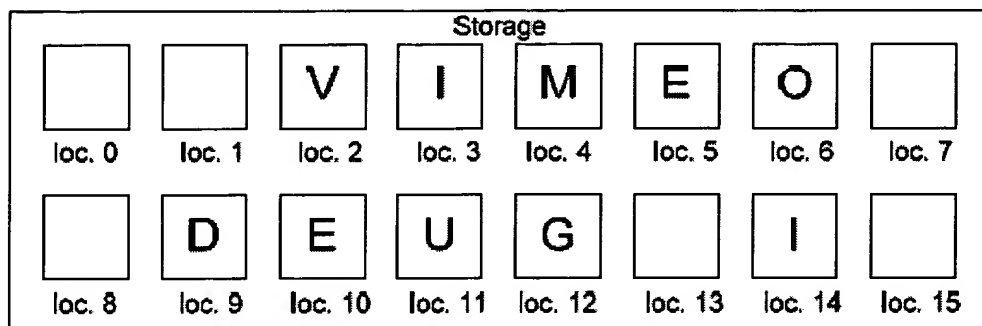
Fig. 1

50

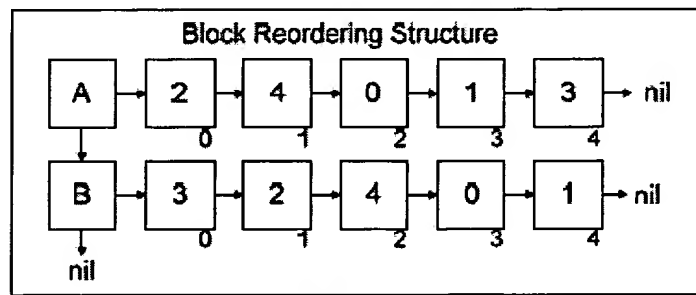
1. An apparatus comprising:
  - a key generator to generate a key according to an identifier value of another apparatus (Figure 1; Specification at page 5, lines 18-19); and
  - a reorderer to reorder blocks of an original content item according to the key (Figure 1; Specification at page 5, lines 19-20).



56. An apparatus comprising:
- a storage to store a copy of a client identifier that identifies more than one client (Figure 1; Specification at page 5, lines 12-17);
  - a key generator to generate a reordering key according to the copy of the client identifier (Figure 1; Specification at page 5, lines 18-19), and
  - a device to enable transfer of a content item to the identified clients in a reordered block format according to the reordering key (Figure 1; Specification at page 5, lines 18-28).



**Fig. 3A**



**Fig. 3B**

79. A method comprising:
- receiving from a first entity, reordered blocks of a content item, the order of said reordered blocks different from the block order for the original content item (Figure 1; Specification at page 5, line 18-page 6, line 2);
  - creating a block reordering structure within a second entity (*See, e.g.*, Figure 1, Figures 3A-3C, Specification at page 6, lines 15-17, page 7, line 23-page 8, line 24); and

accessing a block of the original content item by retrieving it from the reordered content item according to the block reordering structure (*See, e.g.*, Figure 1, Figures 3A-3C, Specification at page 6, lines 15-21, page 7, line 23-page 8, line 24).

91. A recordable medium having recorded thereon a reordered content item resulting from the process comprising:

generating a key in response to an identifier value of a content retrieval entity (Figure 1; Specification at page 5, lines 18-19); and

reordering, as controlled by the key, blocks of an original content item to create the reordered content item (Figure 1; Specification at page 5, lines 19-20).

At this point, no issue has been raised that would suggest that the words in the claims have any meaning other than their ordinary meanings. Nothing in this section should be taken as an indication that any claim term has a meaning other than its ordinary meaning.

**GROUND OF REJECTION TO BE REVIEWED ON APPEAL**

- A. Whether claims 1-6, 10-26, 56-59, 63-76, 79-81, and 91-96 are anticipated under 35 U.S.C. § 102(e) by Etzel (U.S. Patent No. 6,577,734).**
  
- B. Whether claims 7-9, 27-28, and 60-62 are unpatentable under 35 U.S.C. § 103(a) over Etzel (U.S. Patent No. 6,577,734) in view of Donald (U.S. Patent No. 6,415,032).**

## ARGUMENT

**A. Are claims 1-6, 10-26, 56-59, 63-76, 79-81, and 91-96 anticipated under 35 U.S.C. § 102(e) by Etzel (U.S. Patent No. 6,577,734)?**

**1. Claims 1, 5, 6, 13-16, 19, 20, 91, and 92**

The apparatus of claim 1 comprises a key operator to generate a key according to an identifier value of another apparatus. The apparatus also comprises a reorderer to reorder blocks of an original content item according to the key. Etzel does not disclose a reorderer that reorders blocks of content according to the key, and according to the rejection of claim 1, Etzel does not teach the claimed key generator.

**a. Etzel Does not Disclose a Reorderer to Reorder Blocks of an Original Content Item According to the Key.**

For example, the examiner cites to column 3, lines 37-67 and column 4, lines 1-9 as disclosing a reorderer for reordering blocks of an original content item according to the key. *See* Paper No. 20050906, page 3. But Etzel does not disclose reordering *blocks* of an original content item. An original content item may be comprised of blocks. *See, e.g.*, specification, page 7, lines 1-6; FIG. 2A. In an embodiment of the present invention, a block reorderer reorders blocks of content that make up the original content item. Compare FIG 2A with FIG 3A, as one example. The individual blocks may also be encrypted, but this is not necessary. Specification, page 4, lines 17-21. Thus, in some embodiments, the blocks may be both encrypted and reordered whereas in other embodiments the blocks may be reordered without being encrypted. Etzel encrypts video in a conventional manner. Column 4, lines 13-20. Presumably, other encryption in Etzel is done conventionally as well. Because Etzel discloses conventional encryption, he does not teach a reorderer to reorder blocks of an original content item.

In the final Office action, the examiner refused to read limitations of the specification into the claims. Paper No. 20050906, page 2. While the Patent and Trademark Office gives claims their broadest reasonable construction, this construction is in light of the specification. *Phillips v. AWH Corp*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc). As stated by the Federal Circuit in *Phillips*, “the rules of the PTO require that application claims must ‘conform to the invention as set forth in the remainder of the specification and the terms and phrases used in the

claims must find clear support or antecedent basis in the description so that the meaning of the terms in the claims may be ascertainable by reference to the description.’ ” As is explained above, a reorderer can reorder blocks of an original content item and the blocks that make up the content item may or may not also be encrypted. Reference to section B of this Brief should further illuminate that Etzel does not disclose the claimed reorderer. As Etzel teaches only conventional encryption, Etzel does not anticipate. Reversal of the rejections is requested.

**b. The Cited Portions of Etzel Do Not Disclose a Key Generator as Claimed**

To reject claim 1, the examiner cites to Etzel at column 2, lines 40-67 and column 3, lines 4-37 as disclosing a key generator for generating a key according to an identifier value of another apparatus. *See* Paper No. 20050906, page 3. These passages of Etzel simply do not teach a key generator that generates a key according to an identifier value of another apparatus. In particular, none of the keys disclosed in these passages are generated by security module 30 according to an identifier value of another apparatus.

For example, the module 30 generates unique program encryption keys. The program encryption keys are the keys used to encrypt data, such as video data, and the module 30 generates the program encryption keys before video is encrypted. *See* Abstract; column 1, line 53-column 2, line 8; column 2, lines 40-65; column 3, line 49-column 4, line 33. Furthermore, as module 30 generates the program encryption keys, they are known only to the module 30. Column 2, lines 52-57. Because Etzel’s unique program encryption keys are generated by module 30 and as such are known only to the module 30, the module 30 does not disclose the claimed key generator, which generates keys according to an identifier value of another apparatus.

The module 30 may also generate unique encryption keys such as  $S_{local}$  and  $S_{id}$ . These keys are different from the program encryption keys. *See* Abstract; column 1, line 53-column 2, line 8; column 2, lines 40-65; column 3, line 9-column 4, line 67; column 5, lines 8-25. For example,  $S_{local}$  is used to encrypt program encryption keys, which are stored in memory 26 in encrypted form. Column 3, lines 38-65. When a video program is to be encrypted, an encrypted program encryption key is unloaded from the memory 26 and is decrypted by  $S_{local}$ . Column 4, lines 9-20. Because  $S_{local}$  is a unique encryption key that is used to encrypt/decrypt unique program encryption keys, and the unique program encryption keys are used to encrypt video, these two types of keys are not the same key.

That Etzel's program encryption keys and encryption keys are not the same is also evidenced by their different places of storage as well as their different functions. For example,  $S_{local}$  and  $S_{id}$  are stored in a secure memory location in a local memory on module 30 and encrypted program encryption keys are stored in memory 26 of processor 25. Column 3, lines 22-34.

Furthermore, module 30 does not generate  $S_{local}$  and  $S_{id}$  according to an identifier value of another apparatus. For example,  $S_{local}$  and  $S_{id}$  are random numbers that are generated when the module 30 is initially booted up during a testing phase of manufacturing. Column 3, lines 4-32. Because  $S_{local}$  and  $S_{id}$  are generated during the manufacture of the module 30 there is no evidence that the module is connected to another apparatus, other than the testing equipment. Even so, Etzel specifies that  $S_{local}$  and  $S_{id}$  are random numbers. Column 3, lines 9-13. Therefore, module 30 does not generate  $S_{local}$  and  $S_{id}$  according to an identifier value of another apparatus and they are not utilized to encrypt video content.

$S_{common}$  is another key disclosed by Etzel.  $S_{common}$  is generated external to module 30 and is supplied thereto during manufacturing/testing. *Id.* Etzel is silent as to how  $S_{common}$  is generated. *Id.* Accordingly, Etzel does not teach that  $S_{common}$  is generated by the claimed key generator.

Because none of the program encryption keys and unique encryption keys  $S_{local}$  and  $S_{id}$  are generated by Etzel's service module 30 according to an identifier value of another apparatus claim 1 is not anticipated per the examiner's rejection. Reversal of the rejection is requested.

## **2. Claims 65-68**

Claims 65-68 depend from claim 56, which is discussed below. Claim 56 recites a client identifier that identifies more than one client. Claim 65 indicates that the client identifier is a serial number and claim 66 indicates that the client identifier is a random number.

Claims 65-68 are rejected based on the same passages of Etzel as claim 1. Paper No. 20050906, page. Accordingly, the arguments set forth in section A1 of this Brief also apply to claims 65-68. Namely, module 30 generates program encryption keys,  $S_{local}$ , and  $S_{id}$  but none of these keys is generated according to an identifier of another apparatus.

Furthermore, in the case of claims 65-68, the client identifier has to identify *more than one client*. It is simply not seen how the cited portions of Etzel teach such a thing. Therefore, reversal of the rejections is requested.

### 3. Claims 56—59

Claim 56 recites a storage to store a copy of a client identifier *that identifies more than one client*, a key generator to generate a reordering key according to the copy of the client identifier, and a device to enable transfer of *a content item to the identified clients in a reordered block format according to the reordering key*.

As is explained above in section A1 of this Brief, Etzel does not disclose a reorderer to reorder blocks of an original content item. For at least the same reasons, Etzel does not disclose a device to enable transfer a content item in a reordered block format. Additionally, as explained in section A1, none of the program encryption keys,  $S_{\text{local}}$ ,  $S_{\text{id}}$ , or  $S_{\text{common}}$  is generated according to a copy of a client identifier. From those same explanations, it is also clear that these keys do not identify more than one client. Thus, reversal of the rejection is requested.

Module 30 may generate other keys. For example, module 30 forms a public key using  $S_{\text{id}}$ . Column 4, lines 34-52. Similarly, module 50, which is also a head-end security module, also generates its own  $S_{\text{id}}$  and public key. Modules 30 and 50 exchange their public keys and each independently generate a client variable key (CV), which is also referred to as a symmetrical encryption key. Column 4, lines 52-66; column 5, line 35-column 6, line 2. To independently generate CV keys, modules 30 and 50 exponentiate the public key of the remote module. *Id.* There is no indication in Etzel that the public key used to create the CV key on module 30 is a public key for any module other than module 50 and *vice versa*. Thus, neither module 30 nor module 50 generates a key according to a copy of a client identifier that identifies more than one client.

Etzel's module 50 generates another "shared key." In this case, module 50 generates a shared key for *each particular* subscriber terminal. Column 6, lines 6-42; column 7, lines 9-33. Thus, Etzel fails to disclose a client identifier that identifies more than one client and generating a key according to the copy of the client identifier. For at least these reasons, reversal of the rejections is requested.

#### **4. Claims 10-11 and 63**

Claim 10 depends from claim 1 and recites a storage to store *a list of identifier values* of a plurality of such other apparatuses wherein, for different identifier values of two such other apparatuses, the key generator *generates different keys*, and wherein, in response to the different keys, the reorderer imposes different new block orders on the original content item. To reject claims 10-11 and 63, the examiner cites to column 4, lines 9-67 and column 5, lines 1-4 of Etzel. There is nothing in the cited portions of Etzel that teach storing a list of identifier values of a plurality of such other apparatuses.

For example, the columns 4 and 5 generally refer to head end security modules 30 and 50. Both of these modules generate their respective public keys,  $PK_{id}$  at a time of manufacture and exchange them when they initially communicate with one another. Column 4, lines 34-62. To the extent that the public keys may be considered an identifier value, each of the security modules 30 and 50 stores a public key for just the other module. One public key from one other module can hardly be considered a list of identifier values of a plurality of such other apparatuses. For at least this reason, reversal is requested.

Furthermore, from Etzel's discussion it appears that modules 30 and 50 each independently generate just the one CV key. Module 30 uses its CV key to encrypt a program encryption key (which was used to encrypt video content) and module 50 uses its CV key to decrypt the encrypted key received from module 30. Column 4, lines 20-25; column 5, line 60-column 6, line 2. Therefore, only one key (CV) is used to encrypt (on module 30) the various program encryption keys. As the portions of Etzel relied on by the examiner do not disclose a key generator that generates different keys, reversal of the rejections is requested.

#### **5. Claim 12**

Claim 12 depends from claim 10 and recites wherein the list includes a first identifier value for a first such other apparatus, and a second identifier value for both a second and a third such apparatus, wherein the second identifier value is different than the first identifier value. The examiner cites to column 3, lines 49-67 and column 4, lines 1-9 as teaching this paragraph of claim 12. This portion of Etzel however, has nothing to do with identifier values for apparatuses. Rather, this section of Etzel discusses the program encryption keys generated by module 30 that



are used to encrypt a digital video stream. As the examiner has not established *prima facie* anticipation, reversal of the rejection is requested.

**6. Claims 21-26 and 71-76**

Claims 21-26 and 71-76 identify different types of original content. For example, claim 21 recites an electronic programming guide and claim 23 recites a digital gift certificate. As has been previously explained, Etzel's video on demand system encrypts video content using a unique program encryption key that is priorly generated at the head end security module 30 and is only known to the head end security module. Column 2, lines 40-66. Thus, for content items such as video, Etzel does not disclose reordering blocks of an original content item according to a key that was generated by a key generator according to an identifier value of another apparatus. That Etzel may be practiced in conjunction with different types of data including financial data and medical data etc. has no bearing. In other words, this type of content is merely encrypted by a program encryption key generated by some unknown means. For this reason, reconsideration of the rejections is requested.

**7. Claim 79**

Claim 79 recites creating a block reordering structure within a second entity, and accessing a block of the original content item by retrieving it from reordered content item according to the block reordering structure. Dependent claim 80 recites generating a local key within the second entity, in response to which the block reordering structure is created. Dependent claim 81 recites wherein the second entity generates the local key according to the identifier value of the second entity. Thus, dependent claims 80 and 81 indicate how the local key is generated and how the block reordering structure is created.

As is explained above, in section A1 of this Brief, Etzel fails to disclose block reordering; hence, Etzel also does not disclose creating a block reordering structure within a second entity, or accessing a block of an original content item by retrieving it from a reordered content item according to a block reordering structure. Furthermore, a block reordering structure, which is created in response to generating a local key, is not tantamount to a key. In other words, because the dependent claims further define the local key in relation to the block reordering structure,

claim 79, which recites a block reordering structure, means something different. Accordingly, Etzel does not anticipate independent claim 79.

For example, if a subscriber terminal receives encrypted video, a program encryption key decrypts the video. Column 7, line 37-column 8, line 4. Etzel is devoid of further explanation.

Alternatively, if an encrypted program encryption key is received at a subscriber terminal, again all that Etzel describes is using a key to decrypt the encrypted program encryption key. *See* column 7, line 61-column 8, line 4. Because Etzel merely describes a key to decrypt video and a key to decrypt the other key, the keys alone without more do not disclose the claimed block reordering structure. Accordingly, Etzel does not anticipate.

In yet another alternative, if an encrypted program encryption key is received at Etzel's module 50, this key may be decrypted and re-encrypted. Again, Etzel does not describe much with respect to the actual encryption/decryption. Therefore, Etzel does not disclose with sufficient specificity to anticipate, creating a reordering structure much less a *block* reordering structure.

For at least these reasons, reversal of the rejections is requested.

## **8. Claim 80**

Claim 80 recites generating a local key within the second entity, in response to which the block reordering structure is created. As explained above with respect to claim 79, Etzel does not create a block reordering structure in response to generating a local key within either module 215 or module 50. Therefore, reversal of the rejection is requested.

## **9. Claim 81**

Claim 81 depends from claim 80 and recites *the second entity* generates the local key according to *the identifier value of the second entity*. Etzel does not anticipate claim 81

For example, if module 215 is considered the second entity, there is no indication that this module generates a local key according to a value that identifies itself. Generally, module 50 generates a variable session key and the variable session key is unique to the particular module 215 and module 50. In particular, the variable session key is a randomly generated key ( $Pk_v$ ) that emanates from the security module 50. Column 5, lines 1-3 and column 6, lines 6-23. The session key  $Pk_v$  is transmitted to the particular module 215 and module 215 forms a shared

key CV using the session key. Column 6, lines 6-46. Because the session key is per-use and randomly generated, module 215 does not create a shared key based on an identifier value of the module 50. Furthermore, because the shared key CV is formed using the session key, there is no identifier of the module 215.

If module 50 is considered to be the second entity, it too does not generate a local key according to a self-identifier. For example, the key CV that module 50 shares with module 30 uses the private key from module 30 in its creation. *See* column 5, lines 35-57. Thus, the key shared between module 50 and 30 to decrypt the encrypted program encryption key received from module 30 is not formed according to an identifier of module 50.

For these reasons stated above, reversal of the rejection is requested.

#### **10. Claims 93-96 and 17 and 18**

Claim 93 depends from claim 92, which depends from claim 91. Claim 91 recites generating a key in response to an identifier value of a content retrieval entity. Claim 92 recites the process being performed in a server, and the content retrieval entity being one of a plurality of clients connectable to the server. Claim 93 recites the server *creating the respective identifier values of the clients* to be mutually unique. There is nothing in Etzel that teaches a server that creates an identifier value of a client that is used to generate a key to reorder blocks of an original content item.

For example, if module 30 is considered to be the server and the content item a video content item, the video is encrypted using a program encryption key which is a unique key generated by the module 30. Column 2, lines 53-65. Thus, module 30 does not create a client identifier value and generate a key in response to the client identifier value.

Similarly, if the program encryption key is considered to be the content item, this key is encrypted using a symmetrical encryption key CV. But the CV key is generated using a public key provided by module 50. Column 4, line 34-column 5, line 66. Clearly, the CV key is generated using a public key from module 50 and not the public key created by the module 30. Thus, module 30 does not create a client identifier value and generate a key in response to the client identifier value.

If module 50 is considered to be the server and the program encryption key to be the content item, Etzel still does not anticipate. For example, module 50 uses a shared key for a

particular subscriber to encrypt the program encryption key before sending it to the subscriber. Column 6, lines 6-46. To create this shared key, a particular module provides a public key to module 50. Column 6, line 6-column 7, line 45. Thus, module 50 does not create an identifier value of module 215 to generate a shared key.

Because neither module 30 nor module 50 create an identifier value for a client that is used to generate a key to reorder blocks of an original content item, Etzel does not anticipate claims 93-96 and reversal of the rejections is requested.

**B. Are claims 7-9, 27-28, and 60-62 unpatentable under 35 U.S.C. § 103(a) over Etzel (U.S. Patent No. 6,577,734) in view of Donald (U.S. Patent No. 6,415,032)?**

**1. Claims 7, 8, 27, and 60**

Like Etzel, Donald has nothing to do with block reordering and thus does not cure the deficiency of Etzel. For example, Donald explains that in a block cipher, a message is encrypted one block at a time. Column 2, lines 30-35. For example, a block cipher breaks a message into blocks, and encrypts and decrypts the message blocks. Column 7, lines 10-13. Donald discusses two types of block permutations, bit position permutation and bit pattern permutation.

In bit position permutations, each bit in a message block is treated as an object. Column 7, line 55-column 8, line 53. Referring to Figure 1 of Donald, there is only one message block shown at 106. The *bits* comprising the eight-bit message block are permuted to encrypt the message block 108. Thus, Donald's bit permutation deals with encryption *within* a single block and not reordering blocks of a content item.

Similarly, Donald's mutates only one message block in his bit pattern permutation example. Referring to Figure 3, there is one message block 306, which is three bits long. There are eight possible permutations of the three-bit message block but all eight permutations are of the exact same message block. Column 9, lines 17-59. Thus, Donald does not reorder blocks that make up a content item but rather mutates the entire block many different ways.

Accordingly, the combination of Etzel and Donald fails to render claims 7 and 60 obvious. Reversal of the rejection is requested.

## **2. Claims 9 and 62**

Claim 9 depends from claim 1 and recites wherein the reordered blocks include a first reordered block of a first block size and a second reordered block of a second block size which is different from the first block size. As was explained in section B1 above, the message block 106 is eight bits. In another example the block is three bits in length. Column 7, line 56-column 8, line 20. Thus, message blocks may be of different size. But there is nothing in the cited portions of Donald that discloses two blocks of an original content item being reordered with respect to each other. For example, if the three-bit block is a first block of an original content item and the eight-bit block is a second block in the content item, there is nothing in Donald to suggest that the blocks themselves are reordered. Rather, the bits within the individual blocks may be reordered.

Additionally, there is nothing in Donald that says that the three-bit block and the eight-bit block are both within the same original content item. As understood by Donald's explanation, both are separate message blocks. Accordingly, as Donald does not cure the deficiency of Etzel, reversal of the rejection is requested.

## **3. Claim 28**

Claims 27 and 28 both depend from claim 1 and both recite that the apparatus further comprises a storage device. Claim 27 recites that the reorderer reorders blocks of the original content item and stores them to the storage device according to *a logical addressing system of the apparatus*. In contrast, claim 28 recites the reorderer reorders blocks of the original content item by directly manipulating the *physical addresses at which the blocks are stored to the storage device*.

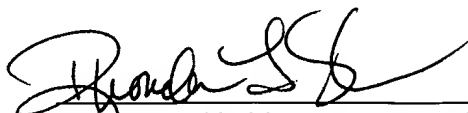
As was explained above, Donald does not disclose reordering blocks. As such, Donald does not disclose reordering blocks of an original content item by directly manipulating the physical addresses as claimed in claim 28. Furthermore, the examiner cites to the exact same portions of Donald to reject both claims 27 and 28. It is respectfully submitted that the examiner cannot have it both ways—the same disclosure cannot teach both logical addressing according to logical addressing of the apparatus and manipulating the physical addresses.

For at least this reason, reversal of the rejection of claim 28 is requested. Primarily, there is no teaching in Donald that the physical addresses in which bits, much less *blocks*, are stored are directly manipulated.

Applicant respectfully requests that each of the final rejections be reversed and that the claims subject to this Appeal be allowed to issue.

Respectfully submitted,

Date: September 20, 2006

A handwritten signature in black ink, appearing to read 'Rhonda L. Sheldon', written over a horizontal line.

Rhonda L. Sheldon, Reg. No. 50,457  
TROP, PRUNER & HU, P.C.  
1616 S. Voss Road, Suite 750  
Houston, TX 77057  
713/468-8880 [Phone]  
713/468-8883 [Fax]

Attorneys for Intel Corporation

## **CLAIMS APPENDIX**

The claims on appeal are:

1. An apparatus comprising:  
a key generator to generate a key according to an identifier value of another apparatus; and  
a reorderer to reorder blocks of an original content item according to the key.
2. The apparatus of claim 1 further comprising:  
a transmitter to distribute the reordered blocks over a wireless broadcast channel.
3. The apparatus of claim 1 further comprising:  
a transmitter to distribute the reordered blocks over a coaxial cable.
4. The apparatus of claim 1 further comprising:  
a transmitter to distribute the reordered blocks over a digital subscriber line (DSL).
5. The apparatus of claim 1 further comprising:  
means for writing the reordered blocks to a removable storage disc.
6. The apparatus of claim 1 further comprising:  
a storage to store the reordered blocks.
7. The apparatus of claim 1 wherein each of the reordered blocks comprises a same data content as its corresponding block from the original content item.
8. The apparatus of claim 1 wherein the reordered blocks are of a uniform block size.

9. The apparatus of claim 1 wherein the reordered blocks include a first reordered block of a first block size and a second reordered block of a second block size which is different than the first block size.

10. The apparatus of claim 1 further comprising:  
a storage to store a list of identifier values of a plurality of such other apparatuses;  
wherein, for different identifier values of two such other apparatuses, the key generator generates different keys; and  
wherein, in response to the different keys, the reorderer imposes different new block orders on the original content item.

11. The apparatus of claim 10 wherein:  
the identifier values in the list are mutually unique; and  
the reorderer imposes a unique new block order on the original content item for each such other apparatus.

12. The apparatus of claim 10 wherein:  
the list includes a first identifier value for a first such other apparatus, and a second identifier value for both a second and a third such other apparatus, wherein the second identifier value is different than the first identifier value; and  
the reorderer imposes a first new block order on the original content item for distribution to the first such other apparatus, and a second, different new block order on the original content item for distribution to either the second or the third such other apparatus.

13. The apparatus of claim 1 wherein the identifier value is a serial number of the other apparatus.

14. The apparatus of claim 1 wherein the identifier value is a random number assigned to the other apparatus.



15. The apparatus of claim 14 wherein the random number has been filtered for primeness and been found to be likely to be prime beyond a predetermined threshold.
16. The apparatus of claim 15 wherein the random number is a prime number.
17. The apparatus of claim 1 wherein:  
the apparatus is a server, the other apparatus is one of a plurality of clients, and  
the server further comprises,  
means for provisioning the clients, including the selection of the identifier values for the clients, and  
means for maintaining a list of the clients' identifier values.
18. The apparatus of claim 1 wherein the identifier value comprises a session key.
19. The apparatus of claim 1 further comprising:  
a transmitter to communicate over a key channel and a content channel.
20. The apparatus of claim 19 wherein the key channel and the content channel are logical channels operating over a same physical medium.
21. The apparatus of claim 1 wherein the original content item comprises an electronic programming guide.
22. The apparatus of claim 1 wherein the original content item comprises ATVEF information.
23. The apparatus of claim 1 wherein the original content item comprises a digital gift certificate.
24. The apparatus of claim 1 wherein the original content item comprises a digital coupon.

25. The apparatus of claim 1 wherein the original content item comprises a movie.
26. The apparatus of claim 1 wherein the original content item comprises an episode of a television show.
27. The apparatus of claim 1 wherein:  
the apparatus further comprises a storage device; and  
the reorderer reorders blocks of the original content item and stores them to the storage device according to a logical addressing system of the apparatus.
28. The apparatus of claim 1 wherein:  
the apparatus further comprises a storage device; and  
the reorderer reorders blocks of the original content item by directly manipulating physical addresses at which the blocks are stored to the storage device.
56. An apparatus comprising:  
a storage to store a copy of a client identifier that identifies more than one client;  
a key generator to generate a reordering key according to the copy of the client identifier, and  
a device to enable transfer of a content item to the identified clients in a reordered block format according to the reordering key.
57. The apparatus of claim 56 wherein the device transfers the reordered block format content item over a wireless broadcast channel.
58. The apparatus of claim 56 wherein the device transfers the reordered block format content item over a coaxial television cable.
59. The apparatus of claim 56 wherein the device transfers the reordered block format content item over a digital subscriber line.

60. The apparatus of claim 56 wherein corresponding respective blocks of the content item in its original block order and reordered block format contain substantially identical data values.

61. The apparatus of claim 56 wherein the content item comprises a plurality of blocks, each of a same block size.

62. The apparatus of claim 56 wherein the content item comprises a plurality of blocks of variable block size.

63. The apparatus of claim 56 wherein:  
the storage maintains a list of respective client identifiers for each of the clients;  
the key generator generates a unique key for each client; and  
for each of the clients receiving the reordered block format content item, a reorderer generates a uniquely reordered block format content item.

64. The apparatus of claim 56 further comprising:  
a key generator to generate at least two reordering keys to reorder an original content item according to the at least to reordering keys.

65. The apparatus of claim 56 wherein the client identifier is a serial number.

66. The apparatus of claim 56 wherein the client identifier is a random number.

67. The apparatus of claim 66 wherein the random number is likely to be prime.

68. The apparatus of claim 66 wherein the random number is prime.

69. The apparatus of claim 56 further comprising:  
a key channel to communicate the key between the clients and the server ; and

a content channel to communicate the content between the server and the clients.

70. The apparatus of claim 69 wherein the key channel and the content channel are logical channels carried over one physical communication medium.

71. The apparatus of claim 56 wherein the content item comprises an electronic programming guide.

72. The apparatus of claim 56 wherein the content item comprises ATVEF information.

73. The apparatus of claim 56 wherein the content item comprises a digital gift certificate.

74. The apparatus of claim 56 wherein the content item comprises an electronic coupon.

75. The apparatus of claim 56 wherein the content item comprises a movie.

76. The apparatus of claim 56 wherein the content item comprises an episode of a television show.

79. A method comprising:  
receiving from a first entity, reordered blocks of a content item, the order of said reordered blocks different from the block order for the original content item;  
creating a block reordering structure within a second entity; and  
accessing a block of the original content item by retrieving it from the reordered content item according to the block reordering structure.

80. The method of claim 79 further comprising:

generating a local key within the second entity, in response to which the block reordering structure is created.

81. The method of claim 80 wherein the second entity generates the local key according to the identifier value of the second entity.

91. A recordable medium having recorded thereon a reordered content item resulting from the process comprising:

generating a key in response to an identifier value of a content retrieval entity; and reordering, as controlled by the key, blocks of an original content item to create the reordered content item.

92. The recordable medium of claim 91 wherein the reordered content item results from the process further comprising:

the process being performed in a server, and the content retrieval entity being one of a plurality of clients connectable to the server; and the server maintaining a list of respective identifier values of the clients.

93. The recordable medium of claim 92 wherein the reordered content item results from the process further comprising:

the server creating the respective identifier values of the clients to be mutually unique.

94. The recordable medium of claim 93 wherein the reordered content item results from the process further comprising:

the server creating the respective identifier values of the clients as serial numbers.

95. The recordable medium of claim 93 wherein the reordered content item results from the process further comprising:

the server creating the respective identifier values of the clients as random numbers.

96. The recordable medium of claim 95 wherein the reordered content item results from the process further comprising:

the server checking the random numbers for at least a threshold likelihood of primeness.

## **EVIDENCE APPENDIX**

None.

## **RELATED PROCEEDINGS APPENDIX**

None.